Support of voice services in IEEE 802.11 wireless LANs

Malathi Veeraraghavan, Nabeel Cocker, Tim Moors
Polytechnic University
{mv@poly.edu, ncocke01@utopia.poly.edu, moors@ieee.org}

Abstract -- **The IEEE 802.11 MAC protocol supports two modes of operation, a random access mode for non-real-time data applications, and a polling mode for real-time applications. We design and analyze a system that uses the polling mode for interactive voice traffic. With larger inter-poll periods, more voice calls can be accommodated, but at the expense of increased delay. For example, our analysis shows that with an inter-poll period of 90 ms, a maximum of 26 voice calls can be handled with a worst-case delay of 303 ms, whereas with an inter-poll period of 60 ms, a maximum of 17 voice calls can be handled with a worst-case delay of 213 ms. We also carry out an error analysis that demonstrates the need for error correction of voice packets.**

I. INTRODUCTION

The IEEE 802.11 wireless LAN [1] is gaining popularity for data applications in campus networks, such as in university campuses and airports. Data rates of these indoor wireless LANs are in the order of 11 Mbps, which is considerably higher than outdoor wireless data services offered through cellular base stations. However, while the use of 802.11 LANs for Internet applications such as web browsing and electronic mail is increasing, there appears to be no immediate commercial interest in using these LANs to carry interactive voice. This is evident in that most commercially-available offerings only implement the 802.11 mode of operation that supports data services, called the Distributed Coordination Function (DCF) mode, and not the second 802.11 mode of operation, designed for real-time services, called the Point Coordination Function (PCF) mode.

The *problem statement* of this work is to determine whether the 802.11 PCF mode is suitable for supporting interactive voice services. This mode of operation uses a polling scheme to provide resource guarantees for real-time sessions. Therefore, more generally, the results of our work are applicable to any polling-based scheme.

The *motivation* for this work comes from an observation that the PCF mode offers a “packet-switched connection-oriented” service, which is well suited for telephony traffic. Telephony traffic has been shown to have alternating periods of talk spurts and silences [2][3]. Packet-switched solutions that take advantage of silences in a given voice call by multiplexing voice data from other calls are more bandwidth-efficient than circuit-switched solutions. This has been one of the primary reasons for the ongoing movement in the telecommunications industry toward moving telephony traffic from DS0 based circuit-switched networks on to packet-switched networks. In wireless networks, where bandwidth is more constrained, the use of packet-switched techniques for carrying voice are indeed needed. While currently most wireless users use cellular or cordless telephones within buildings, the availability of an 802.11 wireless LAN will enable a more efficient usage of overall wireless bandwidth. By having in-building users use wireless LAN access for their voice calls, more of the cellular resources are available for outdoors users and buildings without 802.11 LANs. Thus, our motivation is to take advantage of the packet-switched aspect of 802.11 to support bursty telephony traffic, and thus achieve better overall wireless bandwidth utilization.

The “connection-oriented” aspect of the PCF mode would allow the network to provide delay guarantees necessary for interactive voice. The end-to-end delay requirement for interactive voice is 25ms without echo cancellers, 150ms with echo cancellers for excellent quality voice, and 400ms with echo cancellers for acceptable quality voice [4]. The PCF mode would allow for delay guarantees to be made for voice calls. *In summary, given the packet-switched and connection-oriented aspects of the PCF mode, the problem addressed in this paper is to determine how exactly to use the PCF mode to carry telephony traffic.* Problems associated with the coexistence of DCF and PCF are also addressed.

Section II briefly summarizes the operation of an 802.11 LAN and surveys related work on this topic. Section III describes our proposed solution. The solution is more than simply stating that we use PCF to carry voice. The IEEE 802.11 specification has many options and management-settable parameters. Specific choices of how to operate such a LAN to support voice calls are required. Section IV provides numerical results for the maximum number of voice calls that can be supported and end-to-end delays. While in Sections III and IV we assume error-free transfers on the wireless medium, in Section V, we consider the effect of errors. Section VI presents our conclusions.

II. BACKGROUND AND RELATED WORK

This section provides some background information on 802.11 LANs and reviews prior work on supporting real-time (interactive) traffic using different MAC schemes.

A. Background

An 802.11 LAN can be operated in an ad hoc configuration, i.e., without an Access Point (AP), or in an infrastructure configuration, i.e., with an AP. The AP serves as a MAC layer bridge between wireless stations as well as between wireless and wired stations. The 802.11 standard specifies a MAC protocol (with the DCF and PCF modes), and three physical layer options: Frequency Hopping Spread Spectrum (FHSS), Direct

Sequence Spread Spectrum (DSSS) and InfraRed (IR). The main focus of our work is on the MAC sublayer and particularly the PCF. We therefore provide a review of the MAC layer with an emphasis on the PCF mode.

The *DCF* mode is the fundamental access method of the 802.11 MAC sublayer and is based on Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA). The time period during which the LAN operates in the DCF mode is known as the Contention Period (CP). Access priority to the medium is controlled through the use of InterFrame Spaces (IFS), i.e., the time interval between frames. There are three types of interframe spaces: the Short IFS (SIFS), the Point coordination function IFS (PIFS) and the Distributed coordination function IFS (DIFS). The SIFS is the shortest interval and is used for transmission of acknowledgements, stations responding to polls from the point coordinator (usually the AP) during the PCF mode, and between fragments if an MAC Service Data Unit (MSDU) is fragmented. Transmissions required to wait only a SIFS interval therefore have the highest priority over the medium. The AP uses the PIFS for example, to initiate the CFP. The DIFS is used by stations during the CP. Transmissions required to wait a DIFS interval have the lowest access priority to the medium.

The *PCF* mode provides contention-free frame transfer and the time period in which the LAN operates in the PCF mode is known as the Contention-Free Period (CFP). The AP performs the function of the point coordinator by gaining control of the medium at the beginning of the CFP after sensing the medium to be idle for a PIFS period. During the CFP, stations that are CF-Pollable (can respond to polls) are polled by the AP. On receiving a poll the station transmits its data after a SIFS interval. In order to poll the stations an AP must maintain a polling list, which is implementation dependent. The CFP must alternate with the CP. The sum of the two periods is called the “superframe” and is shown in Fig. 1. It may happen that a station begins to transmit a frame just before the end of the CP, hence elongating the current superframe and shortening the next CFP as shown in Fig. 1. We refer to this as “stretching.”

To understand the effect of *stretching* on the CFP, one should allow for an MSDU of maximum size (which is 2304 bytes) to be sent right before the end of the superframe. If the Fragmentation Threshold (a management-settable parameter) is not equal to this maximum size, then additional overhead will be incurred due to the fragmentation of the MSDU. All fragments of an MSDU are sent SIFS intervals apart, which means that the AP in waiting for a PIFS interval to initiate the CFP cannot acquire the medium in between fragment transmissions. Thus, in the worst case, the stretching period could be as large as is needed to send a 2304 byte payload with fragmentation.

The AP initiates the CFP by transmitting a Beacon frame. If the traffic during the CFP is light and/or the AP has completed polling all the stations on the polling list, it can end the CFP by transmitting a CF-End frame. The contention-free repetition interval (CFPPeriod) is the reciprocal of the rate at which the AP initiates the CFP. The AP then takes control of the medium and starts polling the stations on its polling list. This is shown in Fig. 2.

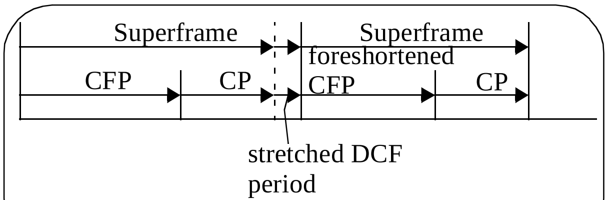

Figure 1: Timing diagram

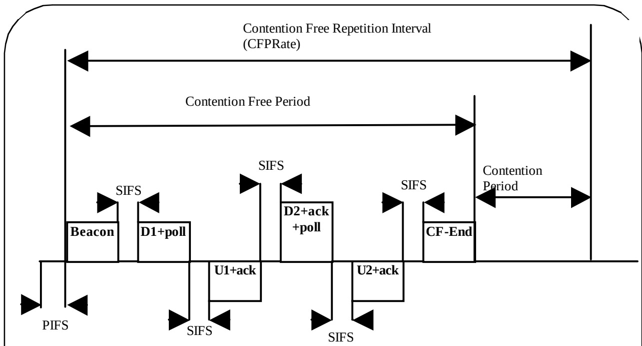

Figure 2: Frame transfers in the PCF mode

Retransmissions are used in 802.11 for error correction both in the DCF and PCF modes. To support error correction, positive acknowledgments (ACKs) are used. These are generated in an interesting manner in 802.11. An ACK for a frame is piggybacked on the next frame even if the latter is not destined to the same station as the sender of the previous frame. There is no mechanism to turn off retransmissions in the PCF mode, or to use different retry counts in the PCF and DCF modes.

Beacons are generated periodically according to the Beacon interval. Mobile stations awaken at listen intervals (which are multiples of beacon intervals) to hear beacons. A beacon is sent at the start of a CFP, but if the CFP duration is larger than the beacon interval then multiple beacons will be sent during a CFP. The CFPPeriod (also a management-settable parameter) indicates the CFP repetition interval. The CFPMaxDuration is also settable and indicated in beacons. For beacons that arrive in the middle of a CFP, the CFPRemainingDuration indicates how long is left in the CFP. Thus, if a mobile station sleeps and awakens on its listen intervals that may not coincide with the start of a CFP, it can still determine the time left in the CFP.

B. *Related work*

MAC protocols can be classified according to whether they assign transmission capability to sources in a fixed manner, at random, or on demand. *Fixed assignment* schemes include time- and frequency-division multiplexing. *Random assignment* schemes are typically used for data traffic. No reservations are made in random assignment. Random assignment schemes (such as 802.11's DCF) tend to offer large and unbounded delays when loads are high. For voice calls, this delay may be acceptable during call setup, but will be less acceptable at the beginning of each talkspurt, or even within a talkspurt. *Demand assignment* schemes may be either centralized or distributed. The assignment may be centralized at a controlling node (as in polling for 802.11 PCF), or in a particular piece of information that circulates amongst nodes (as in token schemes, although these are not widely used in wireless networks because tokens can frequently become lost through bit errors or inaccessible through a station moving out of range). In distributed demand assigned schemes, nodes request access before transmitting, shifting the multiple access problem to the request channel. For example, requests may be sent using either fixed assignment (e.g. using a bit map ([5], pp. 254-5) or random assignment (e.g. PRMA [6]). Reference [7] classifies demand assignment schemes as reservation based schemes, such as DQRUMA [8], polling schemes, or token-ring schemes. Reference [9] makes the case for either prioritization or polling for demand assignment rather than reservation schemes because of packet (burst) access delays associated with the latter.

On the specific topic of how to support voice traffic on MAC protocols, there is a very rich literature. Since we cannot list all these papers, we simply list three survey papers [9]-[11]. More specifically, polling based MAC protocols have been used to support real-time communication in the wireless environment in several protocols other than 802.11 (e.g. [12] and [13]). However, given that 802.11 is now an IEEE standard, papers are focussing on this protocol to support voice. Papers presenting results from simulating the polling used in the PCF mode of IEEE 802.11 include [14]-[16]. In contrast, our paper provides an analysis of polling, and suggests not just how performance varies with the number of calls, but also how 802.11 parameters should be set to support voice calls, and provides details about the polling scheme and method for reconstructing voice timing. Visser and El Zarki [14] consider voice without echo cancellers, and so are forced to use a 20ms superframe which limits the number of admissible voice calls. Crow et al. [15], [16] account for transmission errors, but use a 410ms superframe which consumes most of the end-to-end delay budget for voice in just one hop across the wireless network. Stine and de Veciana [17] also consider voice using PCF, but emphasize the power consumption aspects of the MAC protocol. Finally, [18] proposes a scheme for carrying real-time traffic in an ad hoc 802.11 LAN (i.e., without an AP), while we only consider the case with APs since we expect most calls to be placed from 802.11 phones to wired phones.

III. Proposed solution

This section describes design choices made to carry interactive voice traffic in the PCF mode. Our solution consists of (i) stating our architectural assumption, (ii) defining user plane actions to send voice data, (iii) defining control plane actions, such as Connection Admission Control (CAC) to admit only a limited number of users into the polling list, and (iv) setting Management Information Base (MIB) variables.

A. Architectural assumption

Given our motivation to enable voice calls to be made from 802.11 users to wired, cellular, or Internet telephones, we assume that the network architecture is as shown in Fig. 3.

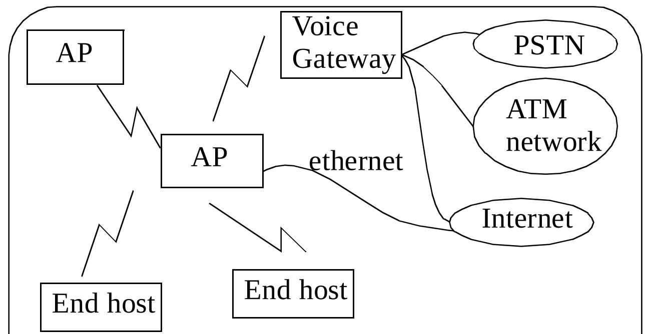

Figure 3: Network architecture

Access points currently have only two interfaces, an 802.11 interface and an ethernet interface. The ethernet interface supports data traffic sent via the DCF mode from mobile stations to the APs. On the other hand, since ethernet does not provide differential Quality of Service (QoS) support, it is not suitable for real-time traffic. Instead, the PCF mode of 802.11 could itself be used from the AP to a voice gateway (see Fig. 3), with the voice gateway supporting interfaces to the PSTN, ATM network (for voice over ATM) and to the Internet (for voice over IP). The voice gateway converts the 802.11 voice protocol stack to PCM voice for use on the PSTN, voice over an ATM Adaptation Layer (AAL) for ATM networks, or voice over Real-time Transport Protocol (RTP)/UDP/IP for IP networks. This architecture allows for end-to-end delay guarantees given that both PSTN and ATM networks are connection-oriented, next-generation IP networks are also expected to support a connection-oriented mode.

In this architecture, by using the PCF mode for voice traffic between an AP and the voice gateway, a voice call to a PSTN/ATM/IP phone uses the same network resources within the 802.11 LAN as an intra-AP call. Thus, for both intra-AP calls and calls between 802.11 users and PSTN/ATM/IP phones, two ends (both sending and receiving) are added to the polling list of an AP. In the case of the latter, the voice gateway extracts voice signals from received 802.11 frames and carries these signals on the protocols used on the wide-area networks.

Extensions to allow for multiple APs on the 802.11 portion of voice calls, and other architectures in which the voice gateway functionality is located in the APs can be considered. In these situations, only one voice call end will be placed on an AP's polling list. These cases are not included in this paper, and will be studied later.

B. User-plane actions

This section describes how voice packets are carried on 802.11. We consider two modes of operation: (i) Constant Bit Rate (CBR) mode in which calls are allocated their peak rate, i.e., as if the sender is always in a talk spurt, and silences are not exploited for other voice calls or data traffic, and (ii) Variable Bit Rate (VBR) mode, in which statistical multiplexing is used and silences in voice calls are used by other voice calls or data traffic.

First consider *whether to send fixed or variable size voice packets*. Given that voice codecs typically produce a given number of bytes every few msec (for example, the Truespeech codec produces 32 bytes every 30ms), one could potentially design the system with fixed size packets. However, the overhead of protocol layers is significant (just the MAC layer header is 34 bytes). Hence we recommend sending all the voice data generated within an interpoll period in one packet. In both modes of operation, CBR and VBR, variable length packets will be created.

The *timing method used for packet voice* needs to be determined. The two choices are Complete Timing Information (CTI) and Null Timing Information (NTI) [19][20]. CTI is used in RTP [21], which uses a relative timestamping method that works well in networks where variable length packets are used and delay variability is high for real-time traffic. The method used in ATM networks to carry voice over AAL2 [22] is NTI. In NTI schemes, each packet does not carry a time indication. Instead if the end-to-end jitter is known, the first packet of a talk spurt is delayed by a build-out delay value equal to the jitter (just in case the first packet arrived with the minimum delay) and then subsequent packets are played out with inter-playout times equal to the packetization intervals. Since ATM cells are fixed size, the packetization delay is fixed and hence even if subsequent cells within a talk spurt experience different delays from the first packet, by playing them out in intervals of T, the packetization delay, the exact sent sequence is reconstructed. So a combination of ATM networks being connection-oriented, which allows for a determination of jitter, and ATM using fixed cell sizes, allows for the use of NTI [22]. In 802.11, even if jitter is known in the use of the PCF mode, the size of packets is not fixed. Given that the exact inter-poll period depends upon whether or not a stretching of the CP occurs, the assumption of a constant inter-packet generation time cannot be made, and hence NTI cannot be used. Therefore, we recommend the use of a CTI scheme.

Since most of the calls starting at 802.11 LAN users can be expected to go outside the LAN, we assume that *all voice packets are sent through the AP* even though in some cases, when both ends are within the coverage area of an AP, voice packets could be sent directly between the ends. In the PCF mode, an AP controls who sends data, but the data could go directly from mobile station to mobile station.

The *protocol stack* used to carry voice frames needs to be specified. Given that delay requirements limit the size of voice packets, we try to avoid as much of the protocol layer overhead as possible. With this goal, we recommend running voice over RTP (given the choice of CTI timing) over LLC/802.11 MAC within the 802.11 LAN (e.g., from an 802.11 endpoint to the voice gateway shown in Fig. 3). The typical RTP stack calls for UDP/IP between RTP and the link layer protocols. However, these layers are not needed for the intra-802.11 LAN portion. The RTP specification [21] states that while UDP/IP is typically used, other lower layer protocols can also be used.

On the question *of how often to poll a voice call*, the AP could traverse its polling list partially per superframe (e.g., poll each node once every other superframe), once per superframe, or multiple times per superframe. In the CBR mode, the AP polls nodes once per superframe. All three options will be considered for the VBR mode, and to support calls with different delay requirements (e.g., calls to ATM/IP phones as shown in Fig. 3 will require shorter delays on the 802.11 portion than calls to PSTN phones or intra-AP calls).

Consider the issue of *when CFPs and CPs terminate*. In the CBR mode, given that each voice call is allocated its peak duration irrespective of the size of the packet, the length of the CP will be determined by the number of voice calls admitted. In the VBR mode, when the AP completes polling all stations in the polling list, it sends a CF-End and starts the CP, even if the CFPMaxDuration is not exhausted (this may occur if several voice users are silent). However, it leaves the medium idle during the CP, even if there are no data users. This choice is made to keep data throughput at acceptable levels.

For both modes, the *superframe length is fixed*. The maximum number of calls that can be admitted is determined by the superframe length. Varying the superframe length would impact delay variations of admitted calls, and hence this is avoided. Interestingly, it is possible to change the superframe length because beacons carry a parameter indicating the superframe length.

The final issue to consider is whether or not voice packets need *error correction*. Our error analysis for voice in Section IV shows that some form of error correction is needed. There are *three* options. *First*, forward error correction (FEC) could be used to correct errors. *Second*, retransmission could be used. In the downstream direction, the AP should resend any packet for which it experiences an ACKTimeout. In the upstream direction, given our assumption that all voice packets (sent in the PCF mode) are routed through the AP, the AP will timeout waiting for a response to its poll, or notice that a received packet is errored, leading it to poll the source mobile station again for a retransmission of the packet. This means that the CAC algorithm should allocate fewer voice calls than the maximum possible if no errors occurred. Delay should be considered carefully when allowing for retransmissions. A *third* option is to deliver errored packets to the signal processor, and have it reconstruct the voice signals.

C. Control plane actions

The 802.11 specification does not describe a method for creating and maintaining the polling list. This is considered out of scope. To ensure that not too many voice calls are admitted to the polling list, we propose a signaling protocol with an associated Connection Admission Control (CAC) procedure. When a mobile user initiates a voice application, a signaling message is sent to the AP (using DCF) requesting to be added to the polling list. The AP maintains the *number of voice calls* admitted to be less than a maximum count. Below we describe how this maximum is determined. The AP then sends a signaling message to the called mobile station (for intra-AP calls) or voice gateway (for PSTN/ATM/IP calls) specifying the reconstruction delay for received voice packets. The *computation of this reconstruction delay* is also explained below. On acceptance of the voice call by the called party/voice gateway, indicated by a response to the AP, the AP places the two 802.11 ends of the voice call on its polling list, and sends a response to the calling end along with a reconstruction delay for this end to use for voice packets it receives from the other end. When a voice call terminates, i.e., when the AP receives a release message, it drops both ends from its polling list. Table 1 lists our notation.

TABLE 1 Parameters

Parameter	Symbol	Value
Maximum duration of the superframe (includes stretching period)	T_{SF}	Varies
Beacon interval in sec	T_b	Varies
Voice coding rate in Kbits/sec	c	8.5
Transmission rate in Mbits/sec	R	2 (FHSS) and 11 (DSSS)
Header overheads (RTP, LLC, MAC with WEP) in bits[a]	h	57×8
Physical layer header size in bits	P	16×8 and 24×8
Maximum number of voice calls in the CBR mode of operation	N_p	Computed
Maximum number of voice calls in the VBR mode of operation	N_s	Computed
Minimum value of the super frame size	T_{SF-min}	Computed
Minimum value of the CP	T_{cp-min}	Computed
Minimum value of the CFP	$T_{cfp-min}$	Computed
Maximum size SDU in bits	S_{maxSDU}	2304×8
Fragment threshold size (payload) in bits	f	1100×8 and 2304×8
Beacon size in bits	B	40×8
CF-end size in bits	CF_{end}	24×8
The SIFS interval	T_{sifs}	0.028ms
A slot time	T_{slot}	0.050ms
Packetization delay to create one minimum sized “sample”	P_{min}	30ms
Time to send a voice packet generated over a superframe duration T_{SF}	T_v	Computed
Time to send an RTS (20 bytes)	T_{rts}	Computed
Time to send a CTS (14 bytes)	T_{cts}	Computed
Time to send an acknowledgment frame without data (14 bytes)	T_{ack}	Computed

TABLE 1 Parameters

a.Reference [1] requires headers, beacons, preambles, etc., to be transmitted at the rate of 1 Mbps to ensure that all stations can listen to these transmissions regardless of their individual data rates.

1) Computation of the maximum number of voice calls

We compute this maximum number for the two modes of operation, CBR and VBR. In the CBR mode, to determine how much time to allocate per call, we need to determine the maximum size of voice packets. The maximum interpoll time in this mode is T_{SF} seconds. Add to this P_{min} to capture the possibility that a voice packetization completes just after a poll. Thus, the largest voice packet size created will be $c(P_{min} + T_{SF})$ bits long. Given that in CBR mode, this time is allocated to each voice call whether or not it generates a packet, the time to handle a voice call (in two directions) is:

$$T_v = \frac{(c \times (P_{min} + T_{SF}) + h + P) \times 4}{R} + 4T_{sifs} \qquad (1)$$

To determine the maximum number of calls, we need to find the minimum duration of the CP, and then use T_{SF} minus this minimum duration for the CFP. T_{cp-min} includes the time to minimally send one frame as specified by Section 9.3.3.3 of [1]. Besides this minimum time, we need to allow for the possibility of stretching as explained in Section II.A. $T_{CP-stretch}$ is the time needed for a maximum size stretch.

After the RTS-CTS exchange with corresponding SIFSs, a maximum size SDU is transmitted in the stretch period as a continuous stream of fragments, without errors or need to backoff. As noted in Section II.A, all fragments and ACKs are sent with only SIFS intervals between them. Each fragment is acknowledged. T_{max} is the time to send a maximum-sized SDU that is fragmented into fragments of size f. To accommodate the maximum number of calls,

$$T_{cp} = T_{cp-min} + T_{cp-stretch}\text{, where} \quad (2)$$

$$T_{cp-min} = 2T_{sifs} + 2T_{slot} + 8T_{ack} + T_{max} \quad (3)$$

$$T_{cp-stretch} = T_{rts} + T_{sifs} + T_{cts} + T_{sifs} + T_{max} \quad (4)$$

$$T_{max} = (m-1)\left(\left\lceil\frac{f+h+P}{R}\right\rceil + T_{ack} + 2T_{sifs}\right) + T_{last} \quad (5)$$

where $m = \lceil S_{maxSDU}/f \rceil$ and T_{last} is given by:

$$T_{last} = \frac{S_{maxSDU} - f(m-1) + h + P}{R} + T_{ack} + 2T_{sifs} \quad (6)$$

To compute the maximum number of voice calls that can be admitted, we divide the time left over for the CFP after allowing for a “stretched” CP and the overhead for beacons and CF-end signals by the time required for one voice call T_v given by (1). In the case of a stretched CP, the CFP is foreshortened and hence the number of beacons is $(T_{SF} - T_{cp})/T_b$. Thus, the maximum number of calls that can be admitted using the CBR mode is

$$N_p = \frac{T_{SF} - T_{cp} - T_{ovhd}}{T_v}\text{, where} \quad (7)$$

$$T_{ovhd} = \left(\frac{B+P}{R} + T_{sifs}\right) \times \left\lfloor \frac{T_{SF} - T_{cp}}{T_b} \right\rfloor + \frac{CF_{end} + P}{R} \quad (8)$$

For the VBR mode of operation, we compute the maximum number of calls for two voice models: Brady’s model [2] and May and Zebo’s [3]. Both of these models are ON-OFF Markov-Modulated Fluid (MMF) models, where in the ON state data is generated at the voice codec rate. The two models differ in the mean holding times of the two states as shown in

TABLE 2 Voice models

Model	Mean ON period	Mean OFF period	p
Brady’s model [2]	1 sec	1.35 sec	0.43
May and Zebo model [3]	352ms	650ms	0.35

Table 2. By admitting more calls than N_p, we are statistically guaranteeing that loss will be less than some number ε. N_s is given by

$$\frac{1}{2pN_s} \sum_{k=2N_p+1}^{2N_s} (k-2N_p)\binom{2N_s}{k} p^k (1-p)^{2N_s-k} \le \varepsilon \quad (9)$$

where p is the probability that a sending end is active. Equation (9) is based on the assumption that if a voice call is not polled in one superframe it is better to drop the packet rather than transmit it on the next superframe due to delay considerations.

2) Computation of build-out delay

As described in Section III.B, we selected the CTI scheme for timing, and RTP for transport. RTP uses relative timestamps. A receiver uses a build-out delay when it reconstructs the voice signal. The build-out delay should be as small as possible. We first determine that the build-out delay should be the maximum possible delay difference between two packets. For example, assume the first packet took d_1 seconds (which

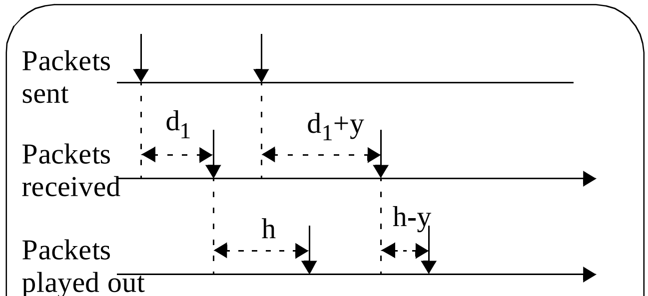

Figure 4: Build-out delay

is unknown) and the total delay d from the start of packetization to delivery at the receiver is within the range $d_{min} \le d \le d_{max}$. How long should this first packet be held at the receiver before playout? Let’s say this is h as shown in Fig. 4. Thus the total delay experienced by the first packet is $d_1 + h$. If the second packet took $d_1 + y$ seconds (where y is known through the relative timestamp), then this packet should be delayed by $h - y$ seconds so that it experiences the same total delay $d_1 + h$ as the first packet. $h - y \ge 0$, which means the smallest value of h is equal to the maximum value of y. The maximum value of y is $d_{max} - d_{min}$, and hence the build-out delay is set to the jitter. Even if $y < 0$, this result holds. In this section, we determine jitter for voice packets in our solution.

To determine jitter, we need to identify how the two 802.11 ends of a voice call are placed on the polling list. The simplest scheme is to have the AP add the two ends to the polling list in sequence as calls arrive. For example, if a call, say A, has two ends A1 and A2, A1 is placed on the polling list immediately followed by A2. The A1 to A2 packets experience short delays because on the A2 poll, data received from A1 can be delivered immediately. However the A2 to A1 packets will experience a greater delay.

In the CBR mode, all calls will have the same delay. We compute the delay in the two directions $k1 \rightarrow k2$, and $k2 \rightarrow k1$ assuming that the $k1$ end is placed on the polling list before the $k2$ end.

$$P_{min} + \frac{T_v}{2} \leq D_{k1 \rightarrow k2} \leq P_{min} + T_{SF} + \frac{T_v}{2}, \text{where} \qquad (10)$$

The best case is that a short packet is created in time P_{min} and packetization completes just before a poll arrives. Given the CBR mode of operation, even if the packet is short, the transmission time allocated per poll and response is $T_v/2$, where T_v is given by (1). Propagation delays are neglected since the radio link is short.

The upper bound is determined by assuming that a poll just misses the creation of a voice packet (P_{min}). The $k1$ end then waits T_{SF} for a poll (this includes the stretching period). On the next poll, when the $k1$ end sends the packet, it is delivered immediately to the $k2$ end. The transmission time is $T_v/2$.

In the opposite direction, $k2 \rightarrow k1$, the delay will be larger. This delay is bounded by

$$P_{min} + T_{SF} - T_{cp-stretch} \leq D_{k2 \rightarrow k1} \leq P_{min} + 2T_{SF} \qquad (11)$$

The lower bound is again determined assuming a small packetization delay. Further, in the best case, there will be no stretching period; in which case, the interpoll time for the $k2$ end is $T_{SF} - T_{cp-stretch}$. The $k1$ end gets polled $T_v/2$ sooner than the $k2$ end on the second poll. This means the time from when the $k2$ end is polled to when the $k1$ end is polled is $T_{SF} - T_{cp-stretch} - T_v/2$. The transmission time adds $T_v/2$.

The upper bound occurs when a poll just misses a packetization (P_{min}). This is followed by a wait of T_{SF} for the next $k2$ poll. This data then waits another $T_{SF} - T_v/2$ time to be delivered to the $k1$ end on the next superframe. The transmission time is $T_v/2$.

Given the jitter values (maximum delay - minimum delay) for the two directions of the voice call, the receiving end in each case can be provided a reconstruction delay by the AP. Thus maximum total delays for the two directions are:

$$TD^{max}_{k1 \rightarrow k2} = D^{max}_{k1 \rightarrow k2} + (D^{max}_{k1 \rightarrow k2} - D^{min}_{k1 \rightarrow k2}) \text{ and}$$

$$TD^{max}_{k2 \rightarrow k1} = D^{max}_{k2 \rightarrow k1} + (D^{max}_{k2 \rightarrow k1} - D^{min}_{k2 \rightarrow k1}) \qquad (12)$$

where the “max” numbers are the upper bounds of (10) and (11), respectively and the “min” numbers are the lower bounds of the same equations.

As calls depart, a call originally scheduled at polling position k can be moved up in the polling list to consolidate all voice calls to the head of the CFP. This would increase the amount of time available for the CP. Build-out delays used by the receiver during the transition will need to be managed by signaling.

For the VBR mode of operation, delay computation is a lot more complex since if a voice call is silent, some other voice call or data packet can take advantage of the silence. This makes the interpoll period highly variable with a possibility of being larger than T_{SF} (unlike in the CBR case, where the interpoll period is a maximum of T_{SF}). Three factors control delay:

(i) value of T_{SF}

(ii) position of the call in the polling list

(iii) whether a call is polled multiple times per superframe, once every superframe, or once every multiple superframes.

In fact, these three methods can be used to offer differential delays for different types of calls. For calls to ATM/IP endpoints, where packetized voice implies higher delays, the 802.11 portion of the call from the wireless user to the AP to the voice gateway needs to be kept small. Algorithms for how to admit calls with differential delays, and to determine reconstructions delays in the VBR mode of operation will be presented in a later paper.

D. *Management plane*

Management plane actions consist of setting MIB variables to enable the operation of the AP and the mobile stations as needed. A few relevant MIB variables are: dot11CFPPeriod, dot11CFPMaxDuration, and dot11BeaconPeriod.

The *dot11CFPPeriod* is the value of the repetition interval (superframe). Section 9.3.3.3 of [1] specifies the minimum CFP period as:

$$T_{cfp-min} = T_B + T_{CF-end} + 3T_{sifs} + (T_v/2), \qquad (13)$$

where $T_v/2$ is the time to send one voice packet and receive a response. Adding this to T_{cp-min} shown in (2), yields

$$T_{SF-min} = T_{cfp-min} + T_{cp-min}. \qquad (14)$$

The superframe duration should be chosen so that $T_{SF} \geq T_{SF-min}$. The CFP repetition interval *dot11CFPPeriod* is advertised to be $T_{SF} - T_{cp-stretch}$, so that the AP can try to gain the medium for the CFP at the end of the CFP repetition interval. In other words, T_{SF} includes the stretching time.

The maximum size of the superframe is a trade-off between the number of calls the network is being engineered for and voice delay constraints. We provide numerical values in the next section.

Having set the CFPPeriod, *dot11CFPMaxDuration*, is set to the CFPPeriod minus T_{cp-min}.

The *dot11BeaconPeriod*, the interval between consecutive beacons transmitted by the AP, is set to equal the *dot11CFPPeriod* variable. This limits the number of beacons generated within a CFP to 1 (i.e., at the start), reducing the Beacon overhead.

IV. NUMERICAL RESULTS

In this section, we determine numerical values for the various parameters set through the management plane (Section A), the maximum call count used in the CAC algorithm at the AP (Section B), and total delay in the CBR mode (Section C).

A. MIB variable numerical values

Table 3 shows the values of certain parameters needed for MIB variable settings (see Section III.D) These are determined for both the 2 Mbps and 11 Mbps data rates, from (3), (4) and (14).

TABLE 3 Values in ms

Data rate (R) Mbps	T_{cp-min}	$T_{cp-stretch}$	T_{SF-min}
2	11.9	10.7	14.9
11	4.4	3.2	5.8

B. Numerical values for maximum number of calls

Fig. 5 shows the maximum number of voice calls that can be accommodated in the CBR mode for 2 Mbps and 11 Mbps at two values of the fragmentation threshold (plots of (7)). For example, with a superframe size of 90 ms, 14 and 26 calls can be admitted on the 2 Mbps and 11 Mbps LANs, respectively. We note that the fragmentation threshold does not have a significant effect on the maximum number of voice calls that can be admitted with the relatively large fragment sizes used in Fig. 5.

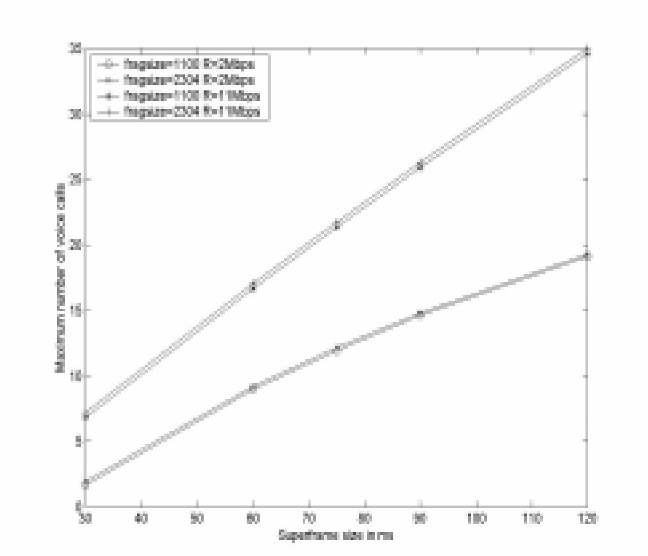

Figure 5: Maximum number of voice calls

Table 4 compares the maximum number of calls admissible in the CBR mode (N_p) and VBR mode (N_s) using both Brady's and May and Zebo's voice models for superframes of 75 and 90 ms. With these superframe sizes, our assumption that a packet should be dropped if not served in a superframe holds. The loss rate, ε, in (9), is assumed to be 10^{-3}. The numbers N_s are optimistic since delays are not considered in (9).

TABLE 4 Maximum number of voice calls
B (Brady's model) and MZ (May and Zebo model)

Tsf (ms)	FH (2 Mbps)			DS (11 Mbps)		
	N_p	N_s (B)	N_s (MZ)	N_p	N_s (B)	N_s (MZ)
75	12	22	27	22	41	51
90	14	26	32	27	52	65

Since the VBR mode exploits silences, the maximum size of a voice packet is larger than $c(T_{SF} + P_{min})$ assumed in (1). We also note that while the maximum number of calls that can be supported in the VBR mode is about double that can be supported in CBR mode, delays will be larger in the VBR mode.

C. Delay results

The maximum delay values determined using (12) are plotted in Fig. 6. Delays for both directions $k1 \rightarrow k2$ and $k2 \rightarrow k1$ at both LAN rates, 2 Mbps and 11 Mbps, are shown. For example, if a superframe size of 90ms is used, then total delays of 121 and 303 msec will be experienced in the $k1 \rightarrow k2$ and $k2 \rightarrow k1$ directions, respectively, for the 802.11 portion of the voice call (on a 11Mbps LAN).

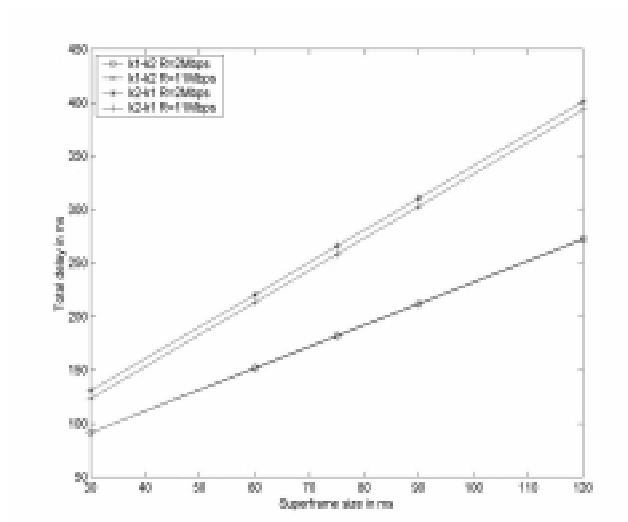

Figure 6: Total delay

V. ERROR ANALYSIS

The 802.11 MAC protocol supports retransmission to handle transmission errors in both PCF and DCF modes. However, retransmissions are typically avoided for real-time traffic due to delay constraints. Here, we examine whether or not some form of error correction is required for voice traffic.

Our analysis takes into account two burst error models. Both are two-state continuous time Markov chains as shown in Fig. 7 [23]. The parameters for the two models are given in Table 5. The first burst error model used to characterize fading is from [23]. The second model is more realistic with higher BERs. The holding times are rough estimates. Indications are that these will be larger, which only improves the probability of the channel holding its state during a packet transmit time.

TABLE 5 Parameters for burst error models

Model	BER_G	BER_B	α	λ
1	10^{-10}	10^{-5}	10/sec	30/sec
2	10^{-4}	10^{-2}	20/sec	10/sec

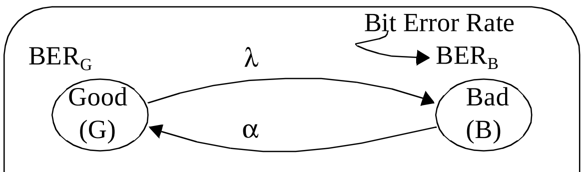

Figure 7: Model of a wireless channel

The time to transmit a voice packet of payload size v bits is given by:

$$T_{v-pkt} = \frac{(v+h+P)}{R}. \tag{15}$$

Three cases are possible:

Case 1: When a voice packet transmission starts, the channel is in the good state, and there is no transition out of this state before packet transmission completes

Case 2: When a voice packet transmission starts, the channel is in the bad state, and there is no transition out of this state before packet transmission completes

Case 3: All other possibilities; packet transmission starts with the channel in either state and the channel undergoes one or more transitions before packet transmission completes

Using the memoryless property of the exponential distribution and by neglecting propagation delays, the probabilities of the three cases can be derived to be:

$$p_{case1} = p_G P(G > T_{v-pkt}) = \frac{\alpha}{\lambda+\alpha} e^{-\lambda T_{v-pkt}} \tag{16}$$

$$p_{case2} = p_B P(B > T_{v-pkt}) = \frac{\lambda}{\lambda+\alpha} e^{-\alpha T_{v-pkt}} \tag{17}$$

$$p_{case3} = 1 - p_{case1} - p_{case2}, \tag{18}$$

where p_G and p_B are the probabilities of starting a packet transmission when the channel is in the good or the bad state, respectively, and are given by:

$$p_G = \frac{\alpha}{\lambda+\alpha} \qquad p_B = \frac{\lambda}{\lambda+\alpha}. \tag{19}$$

The probabilities of a packet error in the three cases are approximated by:

$$\varepsilon_{case1} = 1 - (1 - BER_G)^{(v+h+P)} \tag{20}$$

$$\varepsilon_{case2} = 1 - (1 - BER_B)^{(v+h+P)} \tag{21}$$

$$\varepsilon_{case3} \le \varepsilon_{case2} \tag{22}$$

Combining the probabilities of the three cases, given by (16) to (18), with the probability of packet errors in the three cases, given by (20) to (22), yields the total packet error probability as

$$p_e \le (p_{case1}\varepsilon_{case1} + p_{case2}\varepsilon_{case2} + p_{case3}\varepsilon_{case2}) \tag{23}$$

where a worst-case error rate is assumed if case 3 happens, i.e., that all bits are subject to BER_B.

In the CBR mode, the largest-sized voice packets are:

$$v = c(T_{SF} + P_{min}) \text{ bits}. \tag{24}$$

This upper bound of p_e is plotted against T_{SF} in Fig. 8 for error models 1 and 2. The 11Mbps network experiences a higher packet error rate even though the packet transmission time can be expected to be shorter owing to the higher data rate. This is because DS packets have a larger preamble than FH packets.

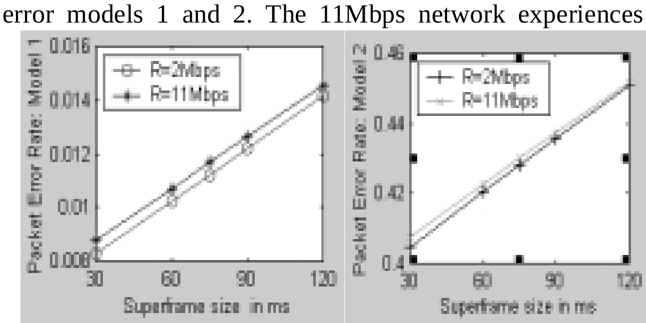

Figure 8: Packet error rates

For a 90 ms T_{SF}, a packet error rate of approximately 10^{-2} and 0.44 for model 1 and model 2, respectively, are observed from the graphs. For voice, with a loss tolerance of 10^{-3}, these error rates are high. This shows a need for error correction. Errors can be handled in one or more of the three ways described in Section III.B.

VI. Conclusions

We demonstrated that the PCF mode of the 802.11 MAC protocol (which uses a polling scheme) can indeed be used to carry telephony traffic. Using a connection admission control algorithm to control the number of voice calls admitted to the polling list, the network can provide delay guarantees. The simplest mode in which to run the LAN during the PCF operation is a Constant Bit Rate (CBR) mode. In this mode, if a voice user is silent, its time is not assigned to any other voice or data user. Ostensibly, this limits the number of calls that can be admitted, but in reality, by limiting delay jitter and hence the maximum delay, the CBR mode allows for a reasonable number of calls to be accommodated. For example, with a 11Mbps 802.11 LAN, 26 voice calls can be admitted if the superframe size (sum of the polling and random-access periods) is 90ms at a maximum delay of 303ms. Also, in this mode, voice calls with different delay requirements (e.g., intra-LAN calls or calls to wired PSTN users vs. calls to Internet phones) can be accommodated by varying the number of times a call is placed on the polling list. Finally, we carried out an error analysis that showed that voice packets can be expected to suffer a high packet error rate. Delay implications of using retransmissions for error correction need to be studied.

ACKNOWLEDGMENTS

We thank Amy Wang, Symbol Tech., Prof. Shiv Panwar, Prof. Henry Bertoni and Ramesh Badri, Polytechnic University, for their help with this work.

REFERENCES

[1] ISO/IEC and IEEE Draft International Standards, "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," ISO/IEC 8802-11, IEEE P802.11/ D10, Jan. 1999.

[2] P. Brady, "A Model for Generating On-Off Speech Patterns in Two-Way Conversation," *Bell Syst. Tech. Journal*, vol. 48, no. 7, pp. 2245-2272, Sept. 1969.

[3] C. E. May and T. J. Zebo, "A summary of speech statistics measured during the TASI-E Rego Park-Ojus field trial," submitted for publication.

[4] ITU-T, "General Characteristics of International Telephone Connections and International Telephone Circuits One-Way Transmission Time" G.114, Feb. 1996.

[5] A. S. Tanenbaum, *Computer Networks*, 3rd ed., Prentice-Hall, 1996.

[6] D. Goodman, R. Valenzuela, K. Gayliard, B. Ramamurthi, "Packet Reservation Multiple Access for local wireless communications", *Proc. 39th IEEE Vehicular Technology Conference*, pp. 701-6, 1988.

[7] A. Leon-Garcia and I. Widjaja, *Communication Networks*, McGraw Hill, 1999.

[8] M. J. Karol, Z. Liu, P. Pancha, "The design and performance of wireless MAC protocols," in *Broadband Wireless Communications*, pp. 225-236, Springer-Verlag, 1998 (papers from the 9th Tyrrhenian Intl. Workshop on Digital Comm., Sept. 1997).

[9] O. Kubbar and H. Mouftah: "Multiple access control protocols for wireless ATM: problems definition and design objectives", *IEEE Comm. Mag.*, vol. 25, no. 11, pp. 93-9, Nov. 1997.

[10] I. Akyildiz, J. McNair, L. Martorell, R. Puigjaner, Y. Yesha, "Medium access control protocols for multimedia traffic in wireless networks," *IEEE Net. Mag.*, vol. 13, no. 4, pp. 39-47, Jul./Aug. 1999.

[11] J. Sanchez, R. Martinez, M. Marcellin, "A survey of MAC protocols proposed for wireless ATM," *IEEE Net. Mag.*, vol. 11, no. 6, pp. 52-62, Nov./Dec. 1997.

[12] S. Alwakeel and M. Al-Fawaz, "DPAP: a dynamic polling based access protocol for wireless networks", *Proc. Ninth IEEE PIMRC*, pp. 1126-30, 1998.

[13] M. Moroney and C. Burkley, "Multiple access protocols for indoor wireless communications", *Proc. IEEE Intl. Conf. on Selected Topics in Wireless Communications*, pp. 406-8, 1992.

[14] M. Visser and M. El Zarki, "Voice and data transmission over an 802.11 wireless network", *Proc. Sixth IEEE International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*, pp. 648-52, Sep. 1995.

[15] B. Crow, I. Widjaja, J. Kim, P. Sakai, "Investigation of the IEEE 802.11 medium access control (MAC) sublayer functions", *Proc. Infocom*, pp. 126-33, Apr. 1997.

[16] B. P Crow, I. Widjaja, J. G. Kim, P. T. Sakai: "IEEE 802.11 Wireless Local Area Networks", *IEEE Comm. Mag.*, vol. 35, no. 9, pp. 116-26, Sep. 1997.

[17] J. Stine and G. de Veciana: "Tactical communications using the IEEE 802.11 MAC protocol", *Proc. IEEE Military Communications Conference (Milcom)*, pp. 575-82, Oct. 1998.

[18] J. L. Sobrinho and A.S. Krishnakumar, "Real-Time Traffic over the IEEE 802.11 Medium Access Control Layer," *Bell Labs Technical Journal*, Autumn 1996.

[19] T. Chen, J. Walrand, D. Messerschmitt, "Dynamic Priority Protocols for Packet Voice," *IEEE JSAC*, vol. 7, no. 1, June. 1989, pp. 632-643.

[20] W. Montgomery, "Techniques for packet voice synchronization," *IEEE JSAC*, vol. 1, no. 6, pp. 1022-8, Dec. 1983.

[21] H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications," IETF RFC 1889, January 1996.

[22] K. Sriram, T. Lyons, Y-T. Wang, "Anomalies due to delay and loss in AAL2 packet voice systems: Performance and Methods of Mitigation," *IEEE JSAC*, vol. 17, no. 1, Jan. 1999, pp. 4-17.

[23] E. Gilbert, "Capacity of a Burst Noise Channel," *Bell Syst. Tech. J.*, vol. 39, Sept. 1960, pp. 1253-66.